



GS/OS

#7: Behavior of SET_DISKSW

Written by: Matt Deatherage

July 1989

This Technical Note discusses changes to the documented behavior of SET_DISKSW in System Software 5.0. This Note is primarily of interest to device driver authors.

GS/OS Reference, Volume 2, states that the system service call SET_DISKSW (\$01FC90) will remove a device's blocks from the cache and place its volumes off line.

With System Software 5.0, this behavior is slightly changed. SET_DISKSW also posts insertion and ejection notices to the GS/OS Notify Procedure queue, so that notification procedures may be called. This requires SET_DISKSW to check the current status of the device to know if the disk switched condition indicates an insertion or an ejection (by comparing the current device status against the device-dispatcher maintained status).

A GS/OS driver may have an interrupt handler present to handle interrupts generated by its device on insertion or ejection (if the hardware is capable of generating such interrupts). Such an interrupt handler will probably want to call SET_DISKSW when an insertion or ejection is detected to make the rest of the operating system aware of it. However, SET_DISKSW obtains the device's status based on the deviceNum and callNum on the GS/OS direct page.

Any driver or interrupt handler calling SET_DISKSW must first save the values for deviceNum and callNum on the GS/OS direct page, replacing callNum with the number of a driver call that accesses media (Apple suggests Driver_Read, \$0002) and replacing deviceNum with the number of the device for which SET_DISKSW is being called. The caller must restore the original values after SET_DISKSW returns.

Although SET_DISKSW saves and restores the GS/OS direct page, the caller must know where the GS/OS direct page is located so it can place the proper parameters there. The value used for the GS/OS direct page should be the value of the D register when the driver receives its Driver_StartUp call. The GS/OS direct page is now guaranteed to remain constant between Driver_StartUp and Driver_ShutDown calls.

Further Reference

- *GS/OS Reference*, Volume 2